IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	**§ Group Art Unit: 2186**
Hofstee, et. al.	**§ Confirmation No.: 9220**
	§ Examiner: Bataille, Pierre M
Serial No.: 10/697,897	**§**
	§ Attorney Docket No.
Filed: October 30, 2003	**§ AUS920030402US1**
	§
Title: System and Method for Sharing Memory by Heterogeneous Processors	**§ IBM Corporation**
	§ Intellectual Property Law
	§ Dept.
	§ 11400 Burnet Road
	§ Austin, Texas 78758

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Certificate of Mailing or Transmission

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 or electronically transmitted to the U.S. Patent and Trademark Office on the date shown below.

/Leslie A. Van Leeuwen, Reg. No. 42,196/ — Feb. 27, 2007
Leslie A. Van Leeuwen, Reg. No. 42,196 — Date

RESPONSE TO OFFICE ACTION

Sir:

A. INTRODUCTORY COMMENTS

In response to the Office Action having a mailing date of November 30, 2006, with a three-month shortened statutory period for response set to expire on February 28, 2007, Applicants respectfully request reconsideration of the outstanding rejections and reexamination of the present application in light of the following remarks and amendments.

No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and the undersigned hereby authorizes the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

B. AMENDMENTS TO THE CLAIMS

1. (Previously Presented) A memory shared by a plurality of heterogeneous processors, comprising:

the shared memory;

wherein the shared memory is accessible by one or more first processors that are adapted to process a first instruction set;

wherein the shared memory is accessible by one or more second processors that are adapted to process a second instruction set; and

a memory map corresponding to the shared memory, wherein the memory map includes cross-references between virtual addresses and real addresses, the memory map and the cross-references shared between the first processors and the second processors.

2. (Canceled)

3. (Previously Presented) The shared memory as described in claim 1 further comprising:

an operating system that operates on one of the first processors, the first processor controlling the memory map.

4. (Original) The shared memory as described in claim 1 wherein each second processor further comprises:

a synergistic processing unit;

a local storage; and

a memory management unit, the memory management unit including a direct memory access controller.

5. (Original) The shared memory as described in claim 4 wherein at least one of the second processors use the direct memory access controller to access the shared memory.

6. (Original) The shared memory as described in claim 4 wherein the local storage is divided into a private storage and a non-private storage.

7. (Original) The shared memory as described in claim 6 wherein the non-private storage is included in the shared memory.

8. (Previously Presented) The shared memory as described in claim 1 wherein the memory map includes a plurality of regions, wherein at least one of the regions is selected from the group consisting of an external system memory region, a local storage aliases region, a TLB region, an MFC region, an operating system region, and an I/O devices region.

9. (Canceled)

10. (Canceled)

11. (Previously Presented) A method for sharing a memory between a plurality of heterogeneous processors, said method comprising:

receiving a memory request;

allocating a first memory partition on the shared memory that corresponds to the memory request, the first memory partition accessible by one or more first processors that are adapted to process a first instruction set;

assigning a second memory partition on the shared memory to one or more second processors that are adapted to process a second instruction set, wherein the first processors and the second processors are heterogeneous;

managing the first memory partition and the second memory partition using a common memory map;

wherein the common memory map includes a plurality of regions, wherein at least one of the regions is selected from the group consisting of an external system memory region, a local storage aliases region, a TLB region, an MFC region, an operating system region, and an I/O devices region; and

wherein the TLB region includes cross-references between virtual addresses and real addresses, the common memory map and the cross-references shared between the first processors and the second processors.

12. (Canceled)

13. (Previously Presented) The method as described in claim 11 wherein one of the first processors includes an operating system whereby the first processor controls the common memory map.

14. (Canceled)

15. (Original) The method as described in claim 11 wherein at least one of the first processors is a Power PC and wherein at least one of the second processors is included in a synergistic processing unit.

16. (Original) The method as described in claim 15 wherein the shared memory corresponds to the synergistic processing unit.

17. (Original) The method as described in claim 11 wherein at least one of the second processors uses a direct memory access controller for accessing the shared memory.

18. (Currently Amended) A computer program product stored on a computer operable media, the computer operable media containing instructions for execution by a computer, which, when executed by the computer, cause the computer to implement a method for sharing a memory between a plurality of heterogeneous processors, the method comprising: ~~for sharing a memory between a plurality of heterogeneous processors, said computer program product comprising:~~

~~means for~~ allocating a first memory partition on the shared memory that corresponds to the memory request, the first memory partition accessible by one or more first processors that are adapted to process a first instruction set;

~~means for~~ assigning a second memory partition on the shared memory to one or more second processors that are adapted to process a second instruction set, wherein the first processors and the second processors are heterogeneous;

~~means for~~ managing the first memory partition and the second memory partition using a common memory map that includes a plurality of regions, wherein at least one of the regions is selected from the group consisting of an external system memory region, a local storage aliases region, a TLB region, an MFC region, an operating system region, and an I/O devices region; and

wherein the TLB region includes cross-references between virtual addresses and real addresses, the common memory map and the cross-references shared between the first processors and the second processors.

19. (Canceled)

20. (Previously Presented) The computer program product as described in claim 18 wherein one of the first processors includes an operating system whereby the first processor controls the common memory map.

21. (Canceled)

22. (Original) The computer program product as described in claim 18 wherein at least one of the first processors is a Power PC and wherein at least one of the second processors is included in a synergistic processing unit.

23. (Original) The computer program product as described in claim 22 wherein the shared memory corresponds to the synergistic processing unit.

24. (Original) The computer program product as described in claim 18 wherein at least one of the second processors uses a direct memory access controller for accessing the shared memory.

25. (Previously Presented) A memory shared by a plurality of heterogeneous processors, comprising:

the memory, wherein the memory includes one or more non-private storage areas, the non-private storage areas

included in one or more second processors that are adapted to process a second instruction set and access the memory;

wherein the shared memory is accessible by one or more first processors that are adapted to process a first instruction set and access the memory; and

a memory map corresponding to the shared memory, wherein the memory map includes cross-references between virtual addresses and real addresses, the memory map and the cross-references shared between the first processors and the second processors.

26. (Original) The shared memory as described in claim 25 wherein each second processor further comprises:

synergistic processing logic which uses private storage, the private storage not included in the shared memory; and

memory management logic for directly accessing the shared memory.

27. (Original) The shared memory as described in claim 25 further comprising:

memory mapping logic that corresponds to the shared memory, wherein the memory mapping logic is shared between the first processors and the second processors.

28. (Original) The shared memory as described in claim 27 further comprising:

an operating system that operates on one of the first processors, the first processor controlling the memory mapping logic.

29. (Original) The shared memory as described in claim 25 wherein one of the first processors configures each of the non-private storage areas.

30. (Previously Added) The shared memory as described in claim 1 further comprising:

wherein the shared memory, the first processors, and the second processors are included on one silicon substrate and are connected using an on chip coherent multi-processor bus.

C. REMARKS

Status of the Claims

Claims 1, 3-8, 11, 13, 15-18, 20, and 22-30 are currently present in the Application, and claims 1, 11, 18, and 25 are independent claims. Claim 18 has been amended, no claims have been canceled, and no claims have been added in this response.

Examiner Interview

Applicants note with appreciation the telephonic interview conducted between Applicants' representative and the Examiner on February 7, 2007. During the telephonic interview, the Examiner and Applicants' representative discussed the 102 reference (Auslander et al., Patent No. 6,601,146). In particular, Applicants' representative discussed that Applicants' invention shares a common memory map between two different **processors** to access a common memory, whereas Auslander discloses a method for sharing a common memory map between two different **processes** that reside on a single processor to access a common memory. Applicants' representative discussed that when Auslander's processes reside on different processors, Auslander uses **different** memory maps for each of the different processors to access the shared memory. No agreement was reached regarding the claims.

Claim Rejections

Claims 1, 3-8, 11, 13, 15-18, 20, and 22-30 stand rejected under 35 U.S.C. § 102 as being anticipated by Auslander et al. (U.S. Patent No. 6,601,146, hereinafter "Auslander"). Applicants respectfully traverse these rejections.

Applicants' claim 1 is directed towards a memory shared by a plurality of heterogeneous processors with limitations comprising:

- the shared memory;
- wherein the shared memory is accessible by one or more first processors that are adapted to process a first instruction set;
- wherein the shared memory is accessible by one or more second processors that are adapted to process a second instruction set; and
- a memory map corresponding to the shared memory, wherein the memory map includes cross-references between virtual addresses and real addresses and is **shared between the first processors and the second processors.**

Claim 1 includes a limitation of a memory map that is "*shared between the first processors and the second processors.*" Applicants' first and second processors use the same architectural address translation mechanism, and thus use the same cross-references between virtual addresses and real addresses, to access a common memory. Applicants' Figure 44A shows that processing unit 4430 and synergistic processing unit 4405 share the same system memory map 4420.

Auslander discloses a method for **processes** to share a common memory map when they reside on a single processor, but never teaches or suggests these processes sharing the same memory map when they reside on different **processors** as claimed by Applicants. Auslander states:

> "This invention provides for an inter-process communication transfer region having a unique physical address, where the region is shared among all processes on **a given processor** of a computer system. This unique physical address is then mapped into a virtual address in the address space of each of the processes. When a first of the processes needs to transfer data to a second of the process, the first

> process stores arguments describing the data in the region. When a second of the processes needs to receive the data, the second process reads the data from the region by **using a second virtual address in its address space** which maps into the unique physical address." (col. 2, line 58 through col. 3, line 2, emphasis added)

Auslander's Figure 2 shows a virtual address space of a client (19a), a virtual address space of a server (19b), and the shared memory (20). Figure 2 shows that the client and the server use different memory maps to access the shared memory, which are mapping 15a (client memory map) and mapping 15b (server memory map). As can be seen, these two memory maps are quite different. First, the virtual address space of mapping 15a (client) locates "Region 1" between "Region 0" and "Region N," whereas the virtual address space of mapping 15b (server) locates "Region 1" between "IPC Region" and "Region' N." Second, mapping 15a's "Region 1" is mapped to a different physical location on shared memory 20 than mapping 15b's "Region' 1." If Auslander did teach that the client and server share the same memory map, mapping 15a and mapping 15b would be identical, which they are not.

During the Examiner interview, the Examiner pointed to an excerpt from Auslander that states:

> "When process A (a client) wishes to communicate with process B (a server), process A makes a (cross address space) procedure call to process B. The data that is communicated is stored as arguments of the function call…" (col. 5, lines 24-26)

Although the excerpt discusses transferring data from a client to a server, the excerpt never teaches or suggests the client and the server sharing the same memory map. As discussed above and shown in Auslander's Figure 2, the client's memory map and the server's memory map are, in fact, different. As such,

Auslander never teaches or suggests "*a memory map corresponding to the shared memory, wherein the memory map includes cross-references between virtual addresses and real addresses and is* ***shared between the first processors and the second processors***" as claimed by Applicants.

Therefore, since Auslander does not teach all of the limitations included in Applicants' claim 1, claim 1 is allowable over Auslander. Independent claim 11 is a method claim including similar limitations of claim 1 and, therefore, is allowable for at least the same reasons as claim 1 is allowable.

Applicants have amended the preamble of independent claim 18 in order to avoid receiving a 101 rejection in subsequent Office communications. Independent claim 18 is a computer program product claim including similar limitations of claim 1 and, therefore, is allowable for at least the same reasons as claim 1 is allowable. Independent claim 25 is a system claim including similar limitations of claim 1 and, therefore, is allowable for at least the same reasons as claim 1 is allowable.

Notwithstanding the fact that claim 30 depends upon claim 1 and, therefore, is allowable for at least the same reasons as claim 1 is allowable, claim 30 adds limitations to claim 1 of:

- wherein the shared memory, the first processors, and the second processors are included on **one silicon substrate** and are connected using an **on chip coherent multi-processor bus.**

Applicants' invention resides on one silicon substrate, and an on chip coherent multi-processor bus connects the first processor, the second processor, and the shared memory. In contrast, Auslander never teaches or suggests such limitation in any part of Auslander. Auslander's Figure 4 shows a single processor 44, but never discloses that two heterogenous

processors, a common memory, and an on chip coherent multi-processor bus reside on one silicon substrate as claimed by Applicants.

Therefore, since Auslander never teaches or suggests all the limitations included in Applicants' claim 30, claim 30 is allowable over Auslander.

Each of the remaining claims 3-8, 13, 15-17, 20, 22-24, and 26-29 depends, either directly or indirectly, upon one of the allowable independent claims 1, 11, 18, or 25. Therefore, claims 3-8, 13, 15-17, 20, 22-24, and 26-29 are allowable for at least the same reasons that their respective independent claims are allowable.

Conclusion

As a result of the foregoing, it is asserted by Applicants that the remaining claims in the Application are in condition for allowance, and Applicants respectfully request an early allowance of such claims.

Applicants respectfully request that the Examiner contact the Applicants' attorney listed below if the Examiner believes that such a discussion would be helpful in resolving any remaining questions or issues related to this Application.

Respectfully submitted,

By /Leslie A. Van Leeuwen, Reg. No. 42,196/
Leslie A. Van Leeuwen, Reg. No. 42,196
Van Leeuwen & Van Leeuwen
Attorney for Applicants
Telephone: (512) 301-6738
Facsimile: (512) 301-6742